Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application

Listing of Claims:

1. (Currently Amended) An asynchronous FIFO circuit comprising:

B9

a memory;

asynchronous reading and writing means of reading a predetermined amount of data from and of writing the predetermined amount of data into said memory on a first-in-first-out basis, the predetermined amount of data including a plurality of words stored in a respective plurality of address locations of the memory;

an error write counter of counting counts up by 1 ~~if the predetermined amount of data~~for each word of the plurality of words containing an error flag that is written into said ~~memory contains an error~~respective plurality of address locations;

an error read counter of counting up by 1 ~~if the predetermined amount of data~~for each word of the plurality of words containing an error flag that is read from said ~~memory contains an error~~respective plurality of address locations;

comparing means of comparing a value of said error write counter with a value of said error read counter, said comparing means outputting a logic level of 0 when the value of said error write counter is coincident with the value of said error read counter, and said comparing means outputting a logic level of 1 if the former value is different from the latter value, wherein the logic level of 1 indicates at least one error flag is set in the plurality of words stored in the respective plurality of address locations.

2. (Currently Amended) An asynchronous FIFO circuit comprising:

a memory having addresses for 2^N words, N being ~~an~~ a positive integer;

B4

a write pointer of counting up by 1 when writing ~~of~~ data into said memory ~~has been~~ is completed, said write pointer counting up by 1 if the memory is not in a full state where the memory is full of data;

a read pointer of counting up by 1 when reading ~~of~~ data from said memory ~~has been~~ is completed, said read pointer counting up by 1 if the memory is not in an empty state where all data has been read from said memory;

a previous read pointer of outputting an output incremented by 1 when reading ~~of~~ data ~~has been~~ is completed, said previous read pointer outputting an output incremented by 1 if said memory is not in the empty state, said previous read pointer always outputting the output of one less value than the value of said read pointer;

an empty flag generating circuit of detecting ~~the~~ an empty state when a value of said write pointer is coincident with a value of said read pointer;

a full flag generating circuit of detecting the empty state of said memory when the value of said write pointer is coincident with a value of said previous read pointer;

a write pointer decoder of decoding the value of said write pointer to generate an address at which data is allowed to be written into said memory;

a data selector of selecting data from an address obtained by decoding the value of said read pointer;

a write flag OR circuit of taking a logic sum of predetermined bits contained in data written into said memory;

a read flag OR circuit of taking a logic sum of predetermined bits contained in data read from said memory;

an error write counter of counting up by 1 if said write flag OR circuit outputs a true logic level when data is written into said memory;

an error read counter of counting up by 1 if said read flag OR circuit outputs a true logic level when data is read from said memory; and

an error comparing circuit of comparing a value of said error write counter with a value of said error read counter to detect whether the value of said error write counter and the value of said error read counter coincide~~,~~

wherein the error comparing circuit outputs a logic level of 0 when the value of said error write counter is coincident with the value of said error read counter, and said error comparing circuit outputs a logic level of 1 if the former value is different from the latter value, and

the logic level of 1 indicates at least one error flag is set in the data stored in the memory.

3. (Original) The asynchronous FIFO circuit according to claim 1 or 2, wherein said error write counter and said error read counter are formed of a gray code counter.

4. (Currently Amended) An asynchronous FIFO data reading and writing method comprising:

an asynchronous reading and writing step of reading a predetermined amount of data from and writing the predetermined amount of data into a memory on a first-in-first-out basis;

an error write ~~counting~~ count step of counting up by 1 if the predetermined amount of data written into said memory contains an error;

an error read ~~counting~~ count step of counting up by 1 if the predetermined amount of data read from said memory contains an error;

B9

a comparing step of comparing a value of said error write ~~counting~~ count step with a value of said error read ~~counting~~ count step, said comparing step outputting a logic level of 0 when the value of said error write ~~counting~~ count step is coincident with the value of said error read ~~counting~~ count step, said comparing step outputting a logic level of 1 if the former value is different from the latter value~~.~~, wherein the logic level of 1 indicates at least one error flag is set in the predetermined amount of data stored in the memory.

5. (Currently Amended) An asynchronous FIFO data reading and writing method comprising:

a write point step of counting up by 1 when writing ~~of~~ data into a memory ~~has been~~ is completed, said memory having addresses for 2^N words, N being ~~an~~ a positive integer, said write point step counting up by 1 if the memory is not in a full state where the memory is full of data;

a read point step of counting up by 1 when reading ~~of~~ data from said memory ~~has been~~ is completed, said read point step counting up by 1 if the memory is not in an empty state where all data has been read from said memory;

a previous read point step of outputting an output incremented by 1 when reading ~~of~~ data from said memory ~~has been~~ is completed, said previous read point step outputting an output incremented by 1 if said memory is not in the empty state, said previous read point step always outputting the output of one less value than said read pointer;

an empty flag generating step of detecting ~~the~~ an empty state when a value of said write pointer is coincident with a value of said read point step;

a full flag generating step of detecting the full state of said memory when the value of said write pointing step is coincident with a value of said previous read pointing step;

a write pointer decode step of decoding the value of said write pointing step to generate an address at which data is allowed to be written into said memory;

a data select step of selecting data from an address obtained by decoding the value of said read pointing step;

B9

a write flag OR step of taking a logic sum of predetermined bits contained in data written into said memory;

a read flag OR step of taking a logic sum of predetermined bits contained in data read from said memory;

an error write count step of counting up by 1 if said write flag OR step outputs a true logic level when data is written into said memory;

an error read count step of counting up by 1 if said read flag OR step outputs a true logic level when data is read from said memory; and

an error comparing step of comparing a value of said error write count step with a value of said error read count step to detect whether the value of said error write count step and the value of said error read count step coincide~~,~~

wherein the error comparing step outputs a logic level of 0 when the value of said error write count step is coincident with the value of said error read count step, and said error comparing step outputs a logic level of 1 if the former value is different from the latter value, and the logic level of 1 indicates at least one error flag is set in the predetermined bits stored in the memory.

6. (Currently Amended) The asynchronous FIFO data reading and writing method according to claim 4 or 5, wherein said error write count step and said error read count step are formed of a gray code count step.